

AUG 17 2007

**AMENDMENTS TO THE CLAIMS**

1. (Original) A method of providing filtered unified logging, the method comprising:  
receiving a message, the message having a predefined severity;  
dropping the message if the message severity does not reach a threshold severity;  
applying one or more filters to the message if the message severity reaches the threshold severity; and  
sending the message to a destination if the message is not filtered out.
2. (Original) The method of claim 1, wherein the message severity and the threshold severity have a severity selected from a group comprising debug, path, info, warning, error, fatal, and none.
3. (Original) The method of claim 1, further including comparing the message severity with the threshold severity.
4. (Original) The method of claim 1, wherein the severity information is inherited.
5. (Original) The method of claim 4, wherein the inherited severity is restricted.
6. (Original) The method of claim 1, wherein as long as the message passes a severity and filter evaluation of a child object, the message is published.
7. (Original) The method of claim 6, wherein the publication is via an inherited log.

8. (Original) The method of claim 1, wherein the method is implemented utilizing Java.
9. (Original) The method of claim 1 wherein the message is sent by an application.
10. (Original) The method of claim 1, wherein the message is selected from a list comprising a log message and a trace message.
11. (Original) A filtered unified logging system, comprising:  
one or more log controllers to represent source data, each of the log controllers receiving one or more messages;  
one or more filters coupled to the log controllers to filter the received messages;  
one or more logs to represent one or more corresponding destinations for the received messages; and  
one or more filters coupled to the logs to filter the messages prior to publication by the logs.
12. (Original) The filtered unified logging system of claim 11, wherein the system is part of Java 2 Enterprise Edition (J2EE) engine.
13. (Original) The filtered unified logging system of claim 11, further including a formatter coupled to each of the logs to determine a format of the received message prior to publication.
14. (Original) The filtered unified logging system of claim 13, wherein the formatter includes one or more subclasses or modules selected from a group

comprising a list formatter, a trace formatter, and an Extensible Markup Language (XML) formatter.

15. (Original) The filtered unified logging system of claim 11, wherein each of the log controllers includes one or more subclasses or modules selected from a group comprising a category and a location.
16. (Original) The filtered unified logging system of claim 11, wherein each of the logs include one or more subclasses or modules selected from a group comprising a stream log, a file log, and a console log.
17. (Original) The filtered unified logging system of claim 16, wherein one or more of the file log and console log are subclasses of the stream log.
18. (Original) The filtered unified logging system of claim 11, wherein the source is an application.
19. (Original) The filtered unified logging system of claim 11, wherein the message includes severity information corresponding to that message.
20. (Original) The filtered unified logging system of claim 19, wherein the severity information includes at least one severity selected from a list comprising debug, path, info, warning, error, fatal, and none.
21. (Original) The filtered unified logging system of claim 19, wherein the severity information is inherited.
22. (Original) The filtered unified logging system of claim 21, wherein the inherited severity is restricted.

23. (Original) The filtered unified logging system of claim 11, wherein the system provides filtered unified logging for both tracing and logging.
24. (Original) The filtered unified logging system of claim 11, wherein the system is implemented using Java.
25. (Original) A computer program for providing filtered unified logging, the computer program comprising:  
a machine readable medium that provides instructions that, if executed by a machine, will cause the machine to perform operations including:  
receiving a message, the message having a predefined severity;  
dropping the message if the message severity does not reach a threshold severity;  
applying one or more filters to the message if the message severity reaches the threshold severity; and  
sending the message to a destination if the message is not filtered out.
26. (Original) The computer program of claim 25, wherein the message severity and the threshold severity have a severity selected from a group comprising debug, path, info, warning, error, fatal, and none.
27. (Original) The computer program of claim 25, wherein the operations further include comparing the message severity with the threshold severity.
28. (Original) The computer program of claim 25, wherein the severity information is inherited.

29. (Original) The computer program of claim 28, wherein the inherited severity

Docket No.: 6570P032  
Application No.: 10/748,012

is restricted.

30. (Original) The computer program of claim 25, wherein as long as the message passes a severity and filter evaluation of a child object, the message is published.
31. (Original) The computer program of claim 30, wherein the publication is via an inherited log.
32. (Original) The computer program of claim 25, wherein one or more of the operations are implemented utilizing Java.
33. (Original) The computer program of claim 25, wherein the message is sent by an application.
34. (Original) The computer program of claim 25, wherein the message is selected from a list comprising a log message and a trace message.